

```
x = c(1, 3, 5, 7, 9, 11) # 創建變數 x 且 x 等於(1, 3, 5, 7, 9, 11)的序列
print(x) # 顯示 x 內容
> [1] 1 3 5 7 9 11
x[2] # 顯示 x 的第 2 個數
> [1] 3
x[-2] # 顯示所有除了第 2 個值以外的值
> [1] 1 5 7 9 11
```

```
seq(from=1, to=7, by=2) # 創建序列
> [1] 1 3 5 7
```

```
rep("yes", times=3) # 創建一個重複的字元
> [1] "yes" "yes" "yes"
```

matrix

```
y = matrix(c(1:9), nrow=3, byrow=TRUE) # 創建變數 y 且等於一個矩陣
>      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
y[1:3] # 顯示 y 的第 1~3 筆資料
> [1] 1 4 7
y[c(1,2), ] # 顯示 y 的第 1, 2 列資料
>      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
y[2:3, 2] # 顯示 y 的第 2, 3 列第 2 行資料
> [1] 5 8
```

```
> ls() # 顯示資料庫變數
[1] "x" "y"
```

```
rm(x) # 將 x 從資料庫移除
rm(list=ls()) # 將資料庫所有變數移除
```

```
help(read.csv) # 幫助了解指令(read.csv)如何編寫
?read.table # 幫助了解指令(read.table)如何編寫
```

read 讀取檔案

```
read.csv(file.choose(), header=T) # 讀取點擊的 csv 檔案
read.table(file.choose(), header=T, sep=",") # 讀取點擊的 csv 檔案
read.delim(file.choose(), header=T) # 讀取點擊的 txt 檔案
read.table(file.choose(), header=T, sep="\t") # 讀取點擊的 txt 檔案
read.table(file="C:/Users/User/Documents/ExportedFileName.txt",
  header=T, sep="\t") # 讀取地址的 txt 檔案
```

write 輸出檔案

```
write.table(DataToExport, file="ExportedFileName.csv", sep=",")
  # 輸出 DataToExport 的 csv 檔(開頭會有編號)
write.table(DataToExport, file="ExportedFileName.csv", row.names=F,
  sep=",") # 輸出 DataToExport 的 csv 檔(開頭沒有編號)
write.table(DataToExport,
  file="C:/Users/User/Documents/ExportedFileName.csv", row.names=F,
  sep=",") # 輸出 DataToExport 的 csv 檔到指定地方(開頭沒有編號)
write.table(DataToExport,
  file="C:/Users/User/Documents/ExportedFileName.txt", row.names=F,
  sep="\t") # 輸出 DataToExport 的 txt 檔到指定地方(開頭沒有編號)
```

```
getwd() # 目前檔案位置
> [1] "C:/Users/User/Desktop/R"
setwd("C:/Users/User/Desktop/") # 前往檔案位置
load("data.Rdata") # 打開檔案(data.Rdata)
load(file.choose()) # 打開所選擇的檔案
save.image("data.Rdata") # 儲存檔案
q() # 儲存檔案
```

安裝 package

```
install.packages() # 安裝選擇的 package
install.packages("epiR") # 安裝 epiR
library(epiR) # 載入 epiR 到工作區
help(package=epiR) # 查詢 epiR 使用方式
remove.packages("epiR") # 移除 epiR
```

data.frame

```
data = data.frame(Col1Name=c("a", "b", "c", "d", "e"),
                  Col2Name=c(1:5), Col3Name=c(6:10))
# 創建變數 data 且等於一個 data.frame
>   Col1Name Col2Name Col3Name
   1      a         1         6
   2      b         2         7
   3      c         3         8
   4      d         4         9
   5      e         5        10
dim(data) # 顯示 data 的維度
> [1] 5 3
head(data) # 顯示 data 前幾個的資料
tail(data) # 顯示 data 最後幾個的資料
names(data) # 顯示 data 橫軸上的名稱
> [1] "Col1Name" "Col2Name" "Col3Name"
data[c(2,3,4), ] # 顯示 x 的第 2~4 筆資料
>   Col1Name Col2Name Col3Name
   2      b         2         7
   3      c         3         8
   4      d         4         9
mean(data$Col2Name) # 算出 data 中名稱是(col2Name)這項的平均
> [1] 3
attach(data) # 將 data 加入到暫存
mean(Col3Name) # 算出 data 中名稱是(Col3Name)這項的平均
> [1] 8
detach(data) # 移除暫存的 data
class(Col1Name) # Col1Name 是屬於什麼類型(numeric, integer, factor,
character)
> [1] "character"
summary(data) # 總結 data 各項的最大最小值、平均等等
>   Col1Name          Col2Name    Col3Name
Length:5          Min.   :1    Min.   : 6
Class :character  1st Qu.:2    1st Qu.: 7
Mode  :character  Median :3    Median : 8
                        Mean  :3    Mean   : 8
                        3rd Qu.:4    3rd Qu.: 9
                        Max.   :5    Max.   :10
```

```
summary(Col2Name)
>   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       2       3       3       4       5
Col2Name = as.factor(Col2Name) # 將 Col2Name 轉變成 factor
levels(Col2Name) # 查看 Col2Name 有哪些 factor
> [1] "1" "2" "3" "4" "5"
Col3Name[Col2Name=="1"] # Col3Name 中 Col2Name 這欄是 1 的資料
> [1] 6
mean(Col3Name[Col2Name=="2"]) # 算出 Col2Name 這欄是 2 的 Col3Name 平均
> [1] 7
Col3Name > 7 # 判斷 Col3Name 資料是否大於 7
> [1] FALSE FALSE TRUE TRUE TRUE
as.numeric(Col3Name > 7) # 判斷 Col3Name 資料是否大於 7，且結果顯示為 0, 1
> [1] 0 0 1 1 1
cbind(Col2Name, Col3Name) # 一起顯示 Col2Name 和 Col3Name
>      Col2Name  Col3Name
[1,]         1         6
[2,]         2         7
[3,]         3         8
[4,]         4         9
[5,]         5        10
```

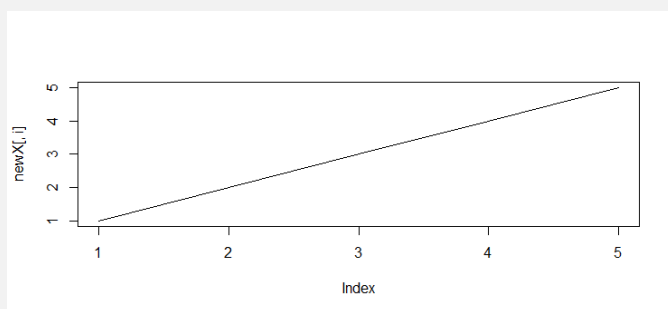
apply 的應用（處理每行數據）

```
Data = array(1:50, c(5, 10))
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    6   11   16   21   26   31   36   41   46
[2,]    2    7   12   17   22   27   32   37   42   47
[3,]    3    8   13   18   23   28   33   38   43   48
[4,]    4    9   14   19   24   29   34   39   44   49
[5,]    5   10   15   20   25   30   35   40   45   50
apply(X=Data, MARGIN=2, FUN=mean, na.rm=T) # 算出每一行的平均
# MARGIN=1 代表列，2 代表行
# FUN 有內建 mean, max, min, sum...，也可自創函數
# na.rm 代表是否自動忽略 na 值(空值)
> [1]  3  8 13 18 23 28 33 38 43 48
colMeans(Data, na.rm=T) # 算出每一行的平均
> [1]  3  8 13 18 23 28 33 38 43 48
```

```

apply(X=Data, MARGIN=2, FUN=quantile, probs=c(0.25, 0.75), na.rm=T)
# 算出每一行的分位數
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 25%      2      7     12     17     22     27     32     37     42     47
 75%      4      9     14     19     24     29     34     39     44     49
apply(X=Data, MARGIN=1, FUN=sum, na.rm=T) #算出每列加總
> [1] 235 245 255 265 275
rowSums(Data, na.rm=T) #算出每列加總
> [1] 235 245 255 265 275
apply(X=Data, MARGIN=2, FUN=plot, type="l", na.rm=T)
# 把每一行畫成直線圖
>

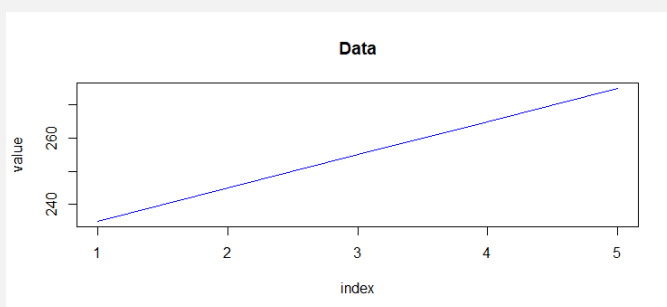
```



```

plot(apply(X=Data, MARGIN=1, FUN=sum, na.rm=T), type="l",
ylab="value", xlab="index", main="Data", col="blue")
#把每一列總和畫成直線圖
>

```



tapply 的應用（處理兩行之間的數據）

```

data = data.frame(Name=c("Mr.a", "Mrs.b", "Mr.c", "Mr.d", "Mrs.e"),
  Gender=as.factor(c("Male", "Female", "Male", "Male", "Female")),
  Weight=c(62, 52, 71, 79, 48))
# 創建變數 data 且等於一個 data.frame
attach(data) # 將 data 加入到暫存

```

```

tapply(X=Weight, INDEX=Gender, FUN=mean, na.rm=T)
# 個別算出每個 Gender 底下 Weight 的平均
  Female      Male
50.00000 70.66667
tapply(X=Weight, INDEX=Gender, FUN=mean, na.rm=T, simplify=FALSE)
$Female
[1] 50

$Male
[1] 70.66667
tapply(X=Weight, INDEX=Gender, FUN=summary, na.rm=T)
$Female
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   48    49     50     50    51     52

$Male
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
62.00  66.50   71.00   70.67   75.00   79.00
tapply(X=Weight, INDEX=Gender, FUN=quantile, probs=c(0.25,0.75),
      na.rm=T)
$Female
25% 75%
 49  51

$Male
25% 75%
66.5 75.0

```